

CLAIMS

1. A method for a primary application to verify the integrity of a secondary application including the steps of:
 - 5 i. obtaining a reference reduced representation by:
 - A. applying a process to obtain schema metadata from the secondary application;
 - B. creating a reference reduced representation of the first obtained schema metadata using an algorithm; and
 - 10 C. storing the reference reduced representation;
 - ii. during execution of the primary application, applying the process to obtain the schema metadata from the secondary application;
 - iii. creating a second reduced representation of the second obtained schema metadata using the algorithm;
 - 15 iv. comparing the reference reduced representation and the second reduced representation; and
 - v. controlling execution of the primary application dependent on the outcome of the comparison.
- 20 2. A method as claimed in claim 1 wherein the secondary application is a database.
3. A method as claimed in claim 2 wherein the schema metadata is selected from the set of tables, columns in tables, datatypes of columns, lengths of
 - 25 columns, custom database data types, foreign keys, constraints, stored procedures, views, triggers, indices, and scheduled jobs.
4. A method as claimed in claim 2 wherein the algorithm is a hash function.
- 30 5. A method as claimed in claim 3 wherein the hash function is one selected from the set of MD5 and CRC32.
6. A method as claimed in claim 2 wherein the algorithm is a lossless compression algorithm.

7. A method as claimed in claim 6 wherein the lossless compression algorithm is one selected from the set of zip, gzip, and bzip2.
- 5 8. A method as claimed in claim 2 wherein the reference reduced representation is stored by embedding the representation within the primary application.
9. A method as claimed in claim 2 wherein the reference reduced representation is stored by embedding the representation within configuration files for the
10 primary application.
10. A method as claimed in claim 2 wherein step (i) is repeated before steps (ii) to (v) at least one time when an expected change occurs to the schema metadata in the database.
- 15 11. A method as claimed in claim 2 wherein the process includes organizing the extracted schema metadata using a nested and determinable method.
12. A method as claimed in claim 11 wherein the nested and determinable
20 method is by alphabetical listing of the schema metadata elements.
13. A method as claimed in claim 11 wherein the nested and determinable method is by default database order of the schema metadata elements.
- 25 14. A method as claimed in claim 11 wherein the nested and determinable method is by creation date order of the schema metadata elements.
15. A method as claimed in claim 11 wherein the nested and determinable method is by table owner of the schema metadata elements.
- 30 16. A method as claimed in claim 2 wherein the execution of the primary application is controlled by halting execution of the primary application.

17. A method as claimed in claim 2 wherein the execution of the primary application is controlled by the primary application sending an error message to one selected from the set of a user of the primary application, a manager of the primary application, a manager of the database, and the database.

5

18. A method as claimed in claim 2 including the step of:
i. requesting a schema stability lock of the database.

19. A method as claimed in claim 2 wherein the process obtains all available schema metadata.

10

20. A method as claimed in claim 2 wherein the process only obtains the schema metadata which would affect the primary application if that schema metadata were to change.

15

21. A method as claimed in claim 2 wherein the process utilizes SQL92 standard to obtain the schema metadata from the database.

22. A method as claimed in claim 2 wherein the process utilizes the database's API to obtain the schema metadata from the database.

20

23. A method as claimed in claim 22 wherein the database's API is JDBC.

24. A system for verifying for a plurality of applications the integrity of one or more databases including:

25

- i. a plurality of applications adapted to store a plurality of previously calculated reduced representations of schema metadata for one or more databases, to extract a plurality of schema metadata from one or more databases, to newly calculate a plurality of reduced representations from the plurality of extracted schema metadata, and to compare each of plurality of previously calculated reduced representations with its corresponding newly calculated reduced representation; and

30

- ii. one or more databases adapted to receive requests for schema metadata from the plurality of applications and to transmit schema metadata to the plurality of applications.

- 5 25. A system as claimed in claim 24 wherein the schema metadata is selected from the set of tables, columns in tables, datatypes of columns, lengths of columns, custom database data types, foreign keys, constraints, stored procedures, views, triggers, indices, and scheduled jobs.
- 10 26. A system as claimed in claim 24 wherein the reduced representations are calculated using a hash function.
- 27. A system as claimed in claim 26 wherein the hash function is one selected from the set of MD5 and CRC32.
- 15 28. A system as claimed in claim 24 wherein reduced representations are calculated using a lossless compression algorithm.
- 20 29. A system as claimed in claim 28 wherein the lossless compression algorithm is one selected from the set of zip, gzip, and bzip2.
- 30. A system as claimed in claim 24 wherein each previously calculated reduced representation is stored by embedding the representation within its associated application.
- 25 31. A system as claimed in claim 24 wherein each previously calculated reduced representation is stored by embedding the representation within configuration files for its associated application.
- 30 32. A system as claimed in claim 24 wherein each schema metadata is organized using a nested and determinable method before its reduced representation is calculated.

33. A system as claimed in claim 32 wherein the nested and determinable method is by alphabetical listing of the schema metadata elements.
- 5 34. A system as claimed in claim 32 wherein the nested and determinable method is by default database order of the schema metadata elements.
35. A system as claimed in claim 32 wherein the nested and determinable method is by creation date order of the schema metadata elements.
- 10 36. A system as claimed in claim 32 wherein the nested and determinable method is by table owner of the schema metadata elements.
37. A system as claimed in claim 24 wherein the result of each comparison controls execution of its associated application
- 15 38. A system as claimed in claim 37 wherein the execution of the application is controlled by halting execution of the application.
39. A system as claimed in claim 37 wherein the execution of the application is controlled by the application sending an error message to one selected from the set of a user of the application, a manager of the application, a manager of the associated database, and the associated database.
- 20 40. A system as claimed in claim 24 wherein the plurality of applications are further adapted to request a schema stability lock of the one or more databases.
- 25 41. A system as claimed in claim 24 wherein each application is adapted to extract all available schema metadata from each database.
- 30 42. A system as claimed in claim 24 wherein each application is adapted to extract the schema metadata which would affect the application if that schema metadata were to change.

43. A system as claimed in claim 24 wherein each application is adapted to utilize SQL92 standard to extract the schema metadata from each database.
44. A system as claimed in claim 24 wherein each application is adapted to utilize the database's API to extract the schema metadata from each database.
45. A system as claimed in claim 44 wherein the database's API is JDBC.
46. A system for verifying for an application the integrity of a database including:
- i. an application;
 - ii. a stored reduced representation of schema metadata of a database; and
 - iii. a verification engine which upon connection to a database obtains a reduced representation of schema metadata from the database and compares it with the stored reduced representation to control the application.
47. A system as claimed in claim 46 wherein the schema metadata is selected from the set of tables, columns in tables, datatypes of columns, lengths of columns, custom database data types, foreign keys, constraints, stored procedures, views, triggers, indices, and scheduled jobs.
48. A system as claimed in claim 46 wherein the reduced representations are calculated using a hash function.
49. A system as claimed in claim 46 wherein the stored reduced representation is stored by embedding the representation within the application.
50. A system as claimed in claim 48 wherein each schema metadata is organized using a nested and determinable method before its reduced representation is calculated.
51. A system as claimed in claim 46 wherein the application is controlled by halting execution of the application.

52. A system as claimed in claim 46 wherein the application is controlled by the application sending an error message to one selected from the set of a user of the application, a manager of the application, a manager of the associated database, and the associated database.
53. A computer system for effecting the method of claim 1.
54. Software for effecting the method of claim 1.
55. Storage media containing software as claimed in claim 54.